IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 15 December 2022, accepted 24 December 2022, date of publication 3 January 2023, date of current version 6 January 2023.

Digital Object Identifier 10.1109/ACCESS.2022.3233921

RESEARCH ARTICLE

Design of Approximate Bilateral Filters for Image Denoising on FPGAs

FANNY SPAGNOLO[1], (Member, IEEE), PASQUALE CORSONELLO[1], (Member, IEEE), FABIO FRUSTACI[1], (Senior Member, IEEE), AND STEFANIA PERRI[2], (Senior Member, IEEE)

[1]Department of Informatics, Modeling, Electronics and Systems Engineering, University of Calabria, 87036 Rende, Italy
[2]Department of Mechanical, Energy and Management Engineering, University of Calabria, 87036 Rende, Italy

Corresponding author: Stefania Perri (s.perri@unical.it)

This work was supported in part by the Programmi Operativi Nazionali (PON) Ricerca and Innovazione-Ministero dell'Università e della Ricerca under Grant 1062_R24_INNOVAZIONE.

ABSTRACT This paper presents the hardware design of fast and low-cost denoising filters suitable to be exploited in the enabling technologies for Industry 5.0. A novel approximate computing strategy is introduced to reduce the computational complexity of the image denoising operation and to comply with real-time requirements. Firstly, it is demonstrated that the novel approximate approach can be helpfully exploited in the design of reconfigurable denoising filters able to reach image qualities as close as possible to the precise software counterparts. The reconfigurability leads to hardware architectures run-time adaptable to different levels of noise, whereas the adopted approximation strategy limits hardware resources and energy requirements. Quality tests, performed at various image and kernel sizes, and noise standard deviations, demonstrate that the approximate denoising approach presented here reaches PSNR and SSIM comparable with the precise denoise filtering. In comparison with state-of-the-art FPGA-based competitors, the novel filters reduce the resources requirements by up to 70%, achieve frame rates up to 35 times higher, and dissipate more than 45% lower power. When implemented within the XC7Z7020 FPGA device, a 5×5 filter designed as proposed here denoises 512×512 grayscale images using only 1689 LUTs, 2635 Flip-Flops and 32 DSPs. Moreover, it processes up to 926.8 frames per second, consumes just 63mW @ 244MHz and, with a noise standard deviation equal to 10, it achieves an average PSNR of ~33dB with an average SSIM of ~0.86.

INDEX TERMS Approximate computing, bilateral filtering, FPGA-based designs, image denoising.

I. INTRODUCTION

The next incoming industrial evolution, Industry 5.0, has the main objective of guaranteeing sustainable and efficient manufacturing solutions by leveraging the collaboration between human experts and intelligent machines [1]. As deeply discussed in [1] and [2], in order to achieve this result, proper enabling technologies are required, such as Artificial Intelligence (AI), Edge Computing (EC), Virtual Reality (VR), Internet of Everything (IoE), Collaborative Robots (Cobots), and many others. These technologies can certainly benefit from fast and reliable computer vision and digital image processing techniques [2], which are suitable to favor the human-machine interaction; to increase the manufacturing efficiency; to monitor the manufacturing processes constantly; and finally, to make accurate decisions in real-time.

Unfortunately, as it is well known, image sensors and acquisition electronics circuits make digital images inherently noisy [3] introducing a noise level often not known a priori. In these cases, noise statistic can be estimated from either a single image or multiple acquired images [4], [5], [6]. Therefore, conceiving methods able to remove or at least reduce the noise effects from acquired images and, at the same time, suitable to adapt this capability to different noise levels is crucial for computer vision and image processing

The associate editor coordinating the review of this manuscript and approving it for publication was Christian Pilato.

techniques in order to obtain the most reliable elaboration results.

Considering that the noise must be removed/reduced without compromising the information captured from the observed scene, total variational regularization (TVR) [7], [8] and bilateral filtering [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19] are among the most popular denoising approaches. The TVR methods have the ability of removing noise with edges preservation, but also with the introduction of artifacts (e.g. blocky, staircasing, etc.). The adaptive models presented in [7] and [8] improve such a behavior by exploiting either eigenanalysis of the structure tensor [7] or a balancing parameter to prioritize noise removal or artifacts elimination [8]. Conversely, bilateral filtering owes the ability of reducing noise while preserving the fundamental information to its significantly complex nonlinear weighted averaging operations performed in both spatial and intensity domains. These operations demand high computational resources, lead to quite low speed performances and cause significant power dissipation. Most of the above-mentioned technologies (e.g. EC and IoE) are being exploited within energy- and resources-constrained environments, in which also achieving high computational speeds is mandatory [1], [2]. Taking these considerations into account, several approximation strategies have been recently proposed [15], [16], [17], [18], [19] to reduce the computational complexity of bilateral filtering. Although these approaches allow achieving interesting trade-offs in terms of image quality and hardware characteristics, most of them [15], [16], [17], [19] are not suitable to adapt themselves at runtime to different noise levels.

This paper presents an innovative approximation strategy for the hardware implementation of fast low-cost denoising filters suitable to be adopted in the enabling technologies for Industry 5.0. The novel approach denoises a digital image by processing each pixel through approximate spatial domain and intensity range kernels, whose coefficients are approximated to unsigned integer values. While the spatial coefficients are approximated depending on the kernel size, the range coefficients are computed by means of a new simple piecewise approximation function, purposely introduced to simplify the nonlinear weighted averaging operations conventionally performed on the pixel intensity. The denoising capability of the proposed approach is analyzed in terms of Peak Signal to Noise Ratio (PSNR) and Structural Similarity (SSIM) [20], [21] referring to image sizes ranging from 256×256 to 1024×1024, kernel sizes varying from 3×3 to 21×21, and zero-mean Gaussian noise with standard deviations ranging between 5 and 60. The proposed approach clearly exhibits its graceful behavior limiting the relative error of PSNR and SSIM, introduced in comparison with the MATLAB software bilateral filtering, to -4% and -14%, respectively, which are up to 12% and 24% better than [15] and [16], while remaining comparable with [17] and [19].

For purposes of comparison with state-of-the-art competitors [15], [16], [17], [18], [19] also in terms of implementation results, several FPGA-based denoising filters designed as proposed here were characterized in terms of utilized Look-Up-Tables (LUTs), Flip-Flops (FFs), Block of RAMs (BRAMs), and Digital Signal Processors (DSPs), speed performances and power consumption, using both the Xilinx Virtex-5 and Zynq-7000 devices families. At a parity of used technology, image and kernel sizes, the proposed designs exhibit speed performances $\times4.36$, $\times35.9$, and $\times4.05$ times higher than [15], [16], and [17], respectively. Moreover, a power dissipation $\times1.85$ lower than [19] is achieved at comparable speed performances and resources requirements. As a further advantage, in comparison with [15], [16], and [17], the designs employing the novel approximation strategy save up to $\sim$70% of utilized LUTs, thus better complying with the requirements of the referred applications.

In comparison with previously published papers, the main contributions of this work are:

1. it presents an innovative approach that reduces the computational complexity of denoising filters by approximating spatial domain and intensity range coefficients to unsigned integers;
2. hardware architectures based on the proposed approach adapt themselves to different noise levels varying at run-time;
3. the novel approximation strategy is extensively characterized in terms of denoising capability and hardware performances, at different operating conditions kernel/image sizes and noise standard deviations: accuracy metrics are evaluated on the entire Miscellaneous USC-SIPI dataset [22] and FPGA-based designs, implemented using the Virtex-5 and the Zynq-7000 devices families, are characterized in terms of resources requirements, speed performances and power consumption.

The rest of the paper is organized as follows. Section II provides a brief background and overviews the state-of-the art. The novel approximation strategy is introduced in Section III. Hardware implementations are detailed in Section IV, whereas accuracy and comparison results are presented in Section V. Finally, conclusions are drawn in Section VI.

II. BACKGROUND AND RELATED WORKS

Bilateral filtering is an efficient local method widely used to process digital images on the basis of both geometric and intensity distances between neighboring pixels [9]. Due to the usage of spatial domain and intensity range kernels, such filters have the ability of reducing noise from an $n\times m$ digital image *Im* while preserving fundamental information, in a way that is tuned to human perception [9]. With *Im(x,y)* being the generic pixel at the coordinate *(x,y)* within the image to be processed, a bilateral filter averages nearby pixels as shown in (1), where: *Om(x,y)* is the output pixel; Ω is the $k\times k$ filter window centered in *Im(x,y)*; *(i,j)* is the coordinate of the neighboring pixels in Ω; and, finally, *Ws(i,j)* and *Wr(i,j)* are

the spatial domain and the intensity range coefficients defined in (2).

$$Om(x,y) = \frac{\sum_{(i,j)\in\Omega} Im(i,j) \times Ws(i,j) \times Wr(i,j)}{\sum_{(i,j)\in\Omega} Ws(i,j) \times Wr(i,j)} \quad (1)$$

$$Ws(i,j) = \exp\left(-\frac{ED(i,j)^2}{2\times\sigma s^2}\right) \quad (2a)$$

$$Wr(i,j) = \exp\left(-\frac{\Delta I(i,j)^2}{2\times\sigma r^2}\right) \quad (2b)$$

It can be seen that *Ws(i,j)* and *Wr(i,j)* exponentially depend, respectively, on the Euclidean Distance $ED(i,j) = \sqrt{(x-i)^2+(y-j)^2}$ between the pixel positions *(x,y)* and *(i,j)*, the intensity difference $\Delta I(i,j) = |Im(x,y) - Im(i,j)|$ and the parameters σs and σr that are typically set on the basis of the kernel size and the noise standard deviation, σ_{noise}, to ensure that $ED(i,j) \leq 3\times\sigma s$ and $\sigma r = 3\times\sigma_{noise}$ [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19].

The above definitions clearly show the computational complexity of bilateral filtering and motivate the recent attempts of introducing efficient approximation strategies [15], [16], [17], [18], [19] to make fast low-cost hardware implementations approachable. The method proposed in [15] considers that, with the spatial domain kernel *Ws* being symmetric and separable, the pixels within Ω can be sorted into groups to be processed by the range filter in parallel, thus allowing high speed performances to be achieved. However, such a solution leads to resources requirements and computational delays building up rapidly with the kernel size. As shown in [16], this effect can be mitigated by approximating the range coefficients through the Taylor expansion. Conversely, the technique presented in [17] conjugates an innovative position-oriented grouping logic with a resource sharing strategy that allows processing the sorted groups of pixels in parallel with a limited resources utilization.

A completely different solution is presented in [18] that replaces the conventional range kernel with a model able to adjust the range coefficients accordingly to the noise estimated at the pixels within Ω. In order to reduce both the computational complexity and the resources requirement as much as possible, the noise-aware kernel obtained in this way is approximated to a binary filter. As an alternative, the efficient piecewise approximation function recently proposed in [19] pre-calculates the exponential functions shown in (2) and selects several approximate fitting points as the kernel coefficients.

For all the above-described approximation strategies, efficient hardware designs are available [15], [16], [17], [18], [19]. They avoid the time expensive computations of the exponential functions, by exploiting LUTs that locally store the approximated values of pre-calculated coefficients. Obviously, the way the latter are approximated leads to a different denoising capability.

TABLE 1. Adopted σs at various kernel sizes.

$k\times k$	Min σs	Adopted σs
3×3	0.5	0.5
5×5	0.95	1
7×7	1.41	2
11×11	2.36	3
21×21	4.71	5

1	4	8	4	1
4	16	32	16	4
8	32	64	32	8
4	16	32	16	4
1	4	8	4	1

(a)

1	2	4	6	4	2	1
2	8	16	24	16	8	2
4	16	32	48	32	16	4
6	24	48	64	48	24	6
4	16	32	48	32	16	4
2	8	16	24	16	8	2
1	2	4	6	4	2	1

(b)

1	2	3	4	6	8	6	4	3	2	1
2	4	8	10	14	16	14	10	8	4	2
3	8	12	20	24	28	24	20	12	8	3
4	10	20	32	40	48	40	32	20	10	4
6	14	24	40	56	60	56	40	24	14	6
8	16	28	48	60	64	60	48	28	16	8
6	14	24	40	56	60	56	40	24	14	6
4	10	20	32	40	48	40	32	20	10	4
3	8	12	20	24	28	24	20	12	8	3
2	4	8	10	14	16	14	10	8	4	2
1	2	3	4	6	8	6	4	3	2	1

(c)

FIGURE 1. Sample approximate spatial range kernels at the size: (a) 5 × 5; (b) 7 × 7; (c) 11 × 11.

III. THE PROPOSED APPROXIMATION TECHNIQUE

The novel method here presented approximates both the spatial domain and the intensity range kernels coefficients to unsigned integers. In the following, the 7-bit word-length is referred to. However, higher bit-widths can be easily introduced if a better quality of filtered images is desired.

A. APPROXIMATING THE SPATIAL DOMAIN KERNEL

In order to compute the possible values of the coefficient *Ws(i,j)* as given in (2a) for several kernel sizes $k\times k$, the parameter σs is set to satisfy the condition $ED(i,j) \leq 3\times\sigma s$ [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19]. Table 1 summarizes the minimum values of σs satisfying the above condition and those adopted here for kernel sizes ranging between 3 × 3 and 21 × 21. The exact values of the spatial kernel coefficients related to these kernel sizes have been scaled by 64. This scaling factor has been chosen to approximate each coefficient *Ws(i,j)* to a 7-bit unsigned integer, thus limiting the loss of information introduced by the approximation and significantly reducing the computational complexity. From Figure 1, that illustrates samples of approximate kernels, it can be seen that the spatial coefficients are approximated only by integer values ranging between 0 and 64 and having at most three asserted bits in their binary representation.

Plots depicted in Figure 2 graphically show how well the adopted approximation fits the real exponential scaled

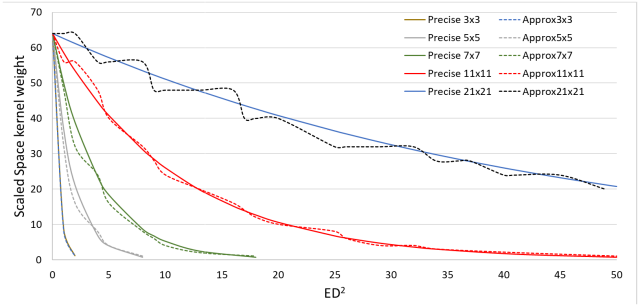

FIGURE 2. Approximated spatial range kernel coefficients at various kernel sizes, compared to the exact exponential functions.

$$
\begin{aligned}
62 \le Wr(i,j) \le 64 &\Rightarrow AWr(i,j)=64\\
58 \le Wr(i,j) < 62 &\Rightarrow AWr(i,j)=60\\
52 \le Wr(i,j) < 58 &\Rightarrow AWr(i,j)=56\\
44 \le Wr(i,j) < 52 &\Rightarrow AWr(i,j)=48\\
36 \le Wr(i,j) < 44 &\Rightarrow AWr(i,j)=40\\
28 \le Wr(i,j) < 36 &\Rightarrow AWr(i,j)=32\\
22 \le Wr(i,j) < 28 &\Rightarrow AWr(i,j)=24\\
18 \le Wr(i,j) < 22 &\Rightarrow AWr(i,j)=20\\
14 \le Wr(i,j) < 18 &\Rightarrow AWr(i,j)=16\\
10 \le Wr(i,j) < 14 &\Rightarrow AWr(i,j)=12\\
6 \le Wr(i,j) < 10 &\Rightarrow AWr(i,j)=8\\
3 \le Wr(i,j) < 6 &\Rightarrow AWr(i,j)=4\\
2 \le Wr(i,j) < 3 &\Rightarrow AWr(i,j)=2\\
1 \le Wr(i,j) < 2 &\Rightarrow AWr(i,j)=1\\
Wr(i,j) < 1 &\Rightarrow AWr(i,j)=0
\end{aligned}
$$

FIGURE 3. Approximation adopted for the intensity range kernel coefficients.

functions at all the referred kernel sizes. To improve the readability of Figure 2, the x-axis range is limited to $ED^2 = 50$.

B. APPROXIMATING THE INTENSITY RANGE KERNEL

Intensity kernel coefficients $Wr(i,j)$ depend on the intensity difference $\Delta I(i,j)$, that, in the case of 8-bit pixels, ranges between 0 and 255, and the parameter σr, set on the basis of the noise standard deviation σ_{noise} [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19].

Again, the exponential function (2b) is scaled by 64 and the 7-bit unsigned integer representation is adopted. Figure 3 reports the simple new rules adopted to approximate $Wr(i,j)$ to its 7-bit version $AWr(i,j)$. In this case, only values having at most four asserted bits in their binary representation are used.

The behavior of the proposed approximation strategy has been analyzed over a relatively wide σ_{noise} range (i.e. 5-75). Figure 4 collects few samples obtained through this investigation. From such experiments, it has been observed that the chosen rules guarantee both a good accuracy and a limited computational complexity in all the examined cases. Moreover, Figure 4 shows the effects of the proposed approximation in comparison with the exact exponential functions at various values of $\sigma_r = 3 \times \sigma_{noise}$. It can be seen that the intensity difference leading to a certain approximate range coefficient $AWr(i,j)$ varies with σr. As an example, when σr = 180, $AWr(i,j)$ is set to 64 for $\Delta I(i,j)$ ranging between 0 and 45. Conversely, with $\sigma r = 60$, the same approximate range coefficient is associated to intensity differences varying from 0 to 15.

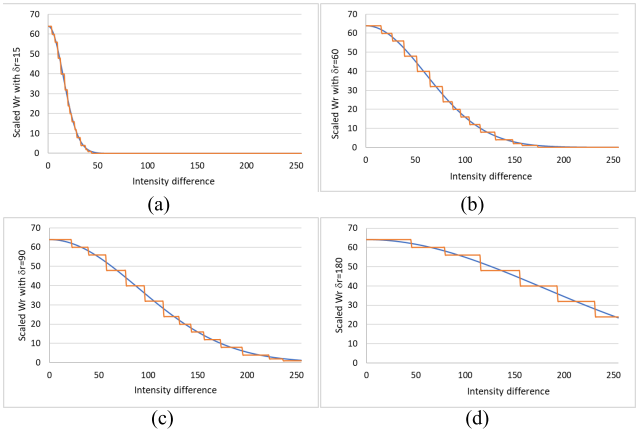

FIGURE 4. Proposed approximation of range kernel coefficient $Wr(i,j)$ at: (a) σr=15; (b) σr=60; (c) σr=90 and (d) σr=180.

$$
AWr(i,j) = \begin{cases}
64 & if\ \Delta(i,j) < 4 \\
60 & if\ \Delta(i,j) < 7 \\
56 & if\ \Delta(i,j) < 10 \\
48 & if\ \Delta(i,j) < 13 \\
40 & if\ \Delta(i,j) < 17 \\
32 & if\ \Delta(i,j) < 20 \\
24 & if\ \Delta(i,j) < 22 \\
20 & if\ \Delta(i,j) < 24 \\
16 & if\ \Delta(i,j) < 27 \\
12 & if\ \Delta(i,j) < 29 \\
8 & if\ \Delta(i,j) < 33 \\
4 & if\ \Delta(i,j) < 38 \\
2 & if\ \Delta(i,j) < 40 \\
1 & if\ \Delta(i,j) < 44 \\
0 & otherwise
\end{cases}
\qquad (a)
$$

$$
AWr(i,j) = \begin{cases}
64 & if\ \Delta(i,j) < 16 \\
60 & if\ \Delta(i,j) < 27 \\
56 & if\ \Delta(i,j) < 39 \\
48 & if\ \Delta(i,j) < 52 \\
40 & if\ \Delta(i,j) < 65 \\
32 & if\ \Delta(i,j) < 78 \\
24 & if\ \Delta(i,j) < 88 \\
20 & if\ \Delta(i,j) < 96 \\
16 & if\ \Delta(i,j) < 105 \\
12 & if\ \Delta(i,j) < 116 \\
8 & if\ \Delta(i,j) < 131 \\
4 & if\ \Delta(i,j) < 149 \\
2 & if\ \Delta(i,j) < 158 \\
1 & if\ \Delta(i,j) < 174 \\
0 & otherwise
\end{cases}
\qquad (b)
$$

$$
AWr(i,j) = \begin{cases}
64 & if\ \Delta(i,j) < 23 \\
60 & if\ \Delta(i,j) < 40 \\
56 & if\ \Delta(i,j) < 58 \\
48 & if\ \Delta(i,j) < 78 \\
40 & if\ \Delta(i,j) < 97 \\
32 & if\ \Delta(i,j) < 116 \\
24 & if\ \Delta(i,j) < 132 \\
20 & if\ \Delta(i,j) < 144 \\
16 & if\ \Delta(i,j) < 157 \\
12 & if\ \Delta(i,j) < 174 \\
8 & if\ \Delta(i,j) < 196 \\
4 & if\ \Delta(i,j) < 223 \\
2 & if\ \Delta(i,j) < 237 \\
1 & otherwise
\end{cases}
\qquad (c)
$$

$$
AWr(i,j) = \begin{cases}
64 & if\ \Delta(i,j) < 46 \\
60 & if\ \Delta(i,j) < 80 \\
56 & if\ \Delta(i,j) < 116 \\
48 & if\ \Delta(i,j) < 156 \\
40 & if\ \Delta(i,j) < 194 \\
32 & if\ \Delta(i,j) < 232 \\
24 & otherwise
\end{cases}
\qquad (d)
$$

FIGURE 5. Piecewise functions obtained with the proposed approximation at: (a) σr=15; (b) σr=60; (c) σr=90 and (d) σr=180.

Figure 5 reports some of the piecewise functions here adopted for different values of σr. It can be seen that, in contrast to existing approximation strategies [15], [17], [19], which find at design time a predefined number of optimal fitting points of the range kernel coefficients at a fixed σr, in the proposed approach also the number of exploited fit-

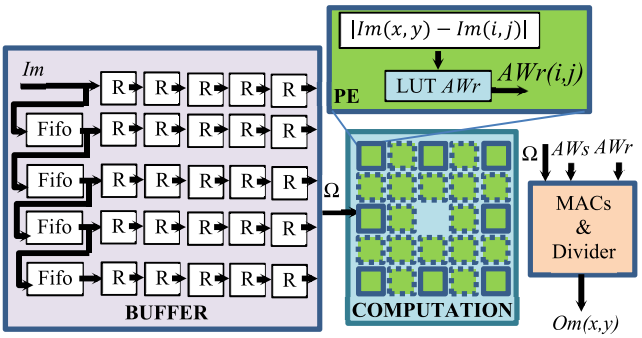

FIGURE 6. Top-level architecture of the proposed design.

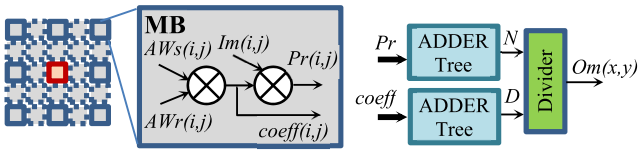

FIGURE 7. Module MACs & Divider.

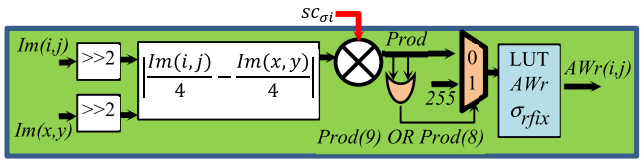

FIGURE 8. Reconfigurable PEs.

ting values changes with σr. It is worth pointing out that this behavior is functional to realize a hardware architecture able to run-time adapt the filtering operations to different noise levels. In order to explain how this is possible, let's suppose that the optimal fitting points complying with the approximation rules of Figure 3 have been found for the predefined $\sigma rfix$. By introducing the scale factor sc_σ, such that $\sigma r = \frac{\sigma rfix}{sc_\sigma}$, equation (2b) can be rewritten as (3), with $sc\Delta I\,(i,j) = sc_\sigma \times \Delta I\,(i,j)$ being the scaled intensity distance.

$$Wr\,(i,j) = \exp\left(-\frac{sc\Delta I(i,j)^2}{2 \times \sigma rfix^2}\right) \tag{3}$$

It is then clear that the approximate *AWr(i,j)* related to σr can be computed using the optimal fitting points precomputed for $\sigma rfix$ by simply scaling the intensity difference $\Delta I(i,j)$ by sc_σ. As an example, let's assume that $\sigma rfix = 60$ and $\sigma r = 180$, that is $sc_\sigma = 1/3$. In this case, an intensity value $\Delta I\,(i,j) = 48$ would be scaled to $sc\Delta I\,(i,j) = 16$ that, through the piecewise function of Figure 5(b), leads to $AWr\,(i,j) = 60$, which is the same approximate range kernel coefficient provided in Figure 5(d) for $\Delta I\,(i,j) = 48$. Analogously, with $\sigma rfix = 60$, $\sigma r = 15$ and $sc_\sigma = 4$, $\Delta I\,(i,j) = 48$ would be scaled to $sc\Delta I\,(i,j) = 192$ that, as ruled in Figure 5(b), leads to $AWr\,(i,j) = 0$, which is the same approximate coefficient provided in Figure 5(a) for $\Delta I\,(i,j) = 48$. It is worth noting that, in order to guarantee a fair use of the scaled intensity distance, $\sigma rfix$ must be chosen so that the corresponding piecewise function employs all the fitting points between 0 and 64. As an example, this means that choosing $\sigma rfix = 180$, as shown in Figure 5(d), would lead to an inefficient solution.

IV. THE NOVEL HARDWARE ARCHITECTURE

Figure 6 depicts the top-level schematic of the basic hardware architecture purposely designed to exploit the above described approximation strategy at a predefined σr. It uses three main modules: i) the BUFFER receives the $n \times m$ input image in the raster scan order and exploits $k-1$ FIFOs, each m-k deep, and $k \times k$ registers to properly arrange the filter window Ω centered at the pixel *Im(x,y)*; ii) the COMPUTATION module processes Ω through $k \times k-1$ Processing Elements (PEs) that implement the approximation rules described in the previous Section to furnish the range kernel *AWr*; iii) the MACs& Divider block receives Ω, *AWr*, the pre-computed approximate spatial domain kernel *AWs* and performs the Multiply Accumulations (MACs) and division operations required to implement (1).

As depicted in the inset of Figure 6, the generic PE computes the absolute difference between the neighboring pixel *Im(i,j)* in Ω and its central pixel *Im(x,y)*. The $\Delta I\,(i,j)$ obtained in this way is then used as the address of the LUT-based memory that stores the possible 7-bit pre-computed values of *AWr(i,j)*. The $k \times k-1$ approximate range kernel coefficients, computed in parallel by as many PEs, are then inputted to the MACs& Divider module.

Figure 7 shows that the latter uses $k \times k-1$ Multiplication Blocks (MBs), each computing the products *Pr(i,j)* and *coeff(i,j)*. It is worth pointing out that, with $k = 5$, *AWs(i,j)* is a power of two constant precomputed at design time. Therefore, the *coeff(i,j)* can be simply computed by left shifts. Furthermore, it must be noted that, since both *AWs(x,y)* and *AWr(x,y)* are always equal to 64, *coeff(x,y)* is constant and *Pr(x,y)* is computed by simply left shifting *Im(x,y)* by 12 bit positions. The $k \times k$ products *Pr* and *coeff* are then accumulated through appropriate Adder Trees that furnish the numerator N and denominator D of (1). Finally, by implementing the non-restoring division algorithm [23], the Divider computes the filtered pixel *Om(x,y)*.

In order to introduce the ability of adapting the denoising filter operations to run-time varying values of σr, the reconfigurable structure illustrated in Figure 8 has been purposely designed for the PEs utilized to approximate the intensity range kernel. First of all, the scale factor sc_σ, such that $\sigma r = \frac{\sigma rfix}{sc_\sigma}$, was introduced. Just as an example, it is assumed that it can vary from 0.25 to 3.75 (i.e. σr can range between $4 \times \sigma rfix$ and $\frac{4}{15} \times \sigma rfix$). To keep the hardware complexity limited, as a proof of concept, sc_σ has been quantized to 4-bit fixed-point values, which, as summarized in Figure 9, can be processed as 4-bit unsigned integers. Consequently, equation (3) has been rewritten as (4) to introduce the integer scale

Binary	sc_σ Fixed Point	$sc_{\sigma i}$ Integer
0 0 0 0	0.0	0
0 0 0 1	0.25	1
0 0 1 0	0.5	2
0 0 1 1	0.75	3
0 1 0 0	1	4
0 1 0 1	1.25	5
0 1 1 0	1.5	6
0 1 1 1	1.75	7
1 0 0 0	2	8
1 0 0 1	2.25	9
1 0 1 0	2.5	10
1 0 1 1	2.75	11
1 1 0 0	3	12
1 1 0 1	3.25	13
1 1 1 0	3.5	14
1 1 1 1	3.75	15

FIGURE 9. Possible representations of sc_σ.

factor $sc_{\sigma i}$.

$$\begin{aligned} Wr\,(i,j) &= \exp\left(-\frac{sc_\sigma \times \Delta I\,(i,j)^2}{2\times \sigma rfix^2}\right) \\ &= \exp\left(-\frac{\frac{sc_{\sigma i}}{4} \times \Delta I\,(i,j)^2}{2\times \sigma rfix^2}\right) \\ &= \exp\left(-\frac{sc_{\sigma i} \times \left|\frac{Im(x,y)}{4} - \frac{Im(i,j)}{4}\right|^2}{2\times \sigma rfix^2}\right) \end{aligned} \quad (4)$$

It is easy to understand that, with *Im(i,j)* and *Im(x,y)* being 8-bit pixels, the absolute difference $\left|\frac{Im(x,y)}{4} - \frac{Im(i,j)}{4}\right|$ is a 6-bit unsigned number and, consequently, the product $Prod = sc_{\sigma i} \times \left|\frac{Im(x,y)}{4} - \frac{Im(i,j)}{4}\right|$ is a 10-bit unsigned number. However, since it represents the intensity difference $\Delta I\,(i,j)$ related to a certain σr, values greater than 255 are saturated. As illustrated in Figure 8, this condition is simply managed by OR-ing the two most significant bits of *Prod*. The subsequent multiplexer then selects either the saturation value 255 or the 8-bit sub-word *Prod(7:0)* as the address of the LUT-based memory that stores the possible 7-bit pre-computed values of *AWr(i,j)* related to the predefined $\sigma rfix$.

V. EXPERIMENTAL RESULTS

A. QUALITY RESULTS

Quality tests on the proposed architecture have been performed using the Miscellaneous USC-SIPI dataset [22] that collects benchmark images with sizes ranging from 256 × 256 to 1024 × 1024. As in related works, the original images were corrupted with additive zero-mean Gaussian noise with standard deviations σ_{noise} ranging between 5 and 60. Then, denoising filters, with kernel sizes varying from 3 × 3 to 21 × 21, have been applied. For the novel approach, the variable noise levels have been managed referring to $\sigma rfix = 60$. Observing the behavior at different kernel sizes is interesting, since establishing which is the best one to adopt is not trivial for any denoising technique, including the accurate software denoising filters provided by MATLAB and here referenced as the touchstone.

TABLE 2. Average PSNR and SSIM.

$\sigma s/\sigma r$	Noisy Images PSNR(dB)/SSIM	MATLAB PSNR(dB)/SSIM	New Filters PSNR(dB)/SSIM
0.5/15		36.01 / 0.9107	35.91 / 0.9082
1/15		37.02 / 0.9358	37.1 / 0.9367
2/15	34.22 / 0.8647	36.14 / 0.9208	36.71 / 0.9307
3/15		35.54 / 0.9126	35.92 / 0.9176
5/15		34.87 / 0.9081	34.97 / 0.9092
0.5/60		25.02 / 0.5128	24.85 /0.5050
1/60		29.13 / 0.7305	28.74 / 0.7058
2/60	22.32 / 03979	29.27 /0.7724	29.51 / 0.7661
3/60		28.46 /0.7514	28.89 / 0.7638
5/60		27.24 /0.7211	27.23 / 0.7215
0.5/90		21.74 / 0.3673	21.56 / 0.3606
1/90		26.53 / 0.6088	25.98 / 0.5738
2/90	18.92 / 0.2683	27.39 / 0.7013	27.21 / 0.6683
3/90		26.72 / 0.6879	26.91 / 0.6921
5/90		25.47 / 0.6529	25.17 / 0.6451

Table 2 summarizes some of the average PSNR and SSIM reached with respect to the original noiseless images by applying the novel approach and the accurate software denoising filters provided by MATLAB. To highlight the denoising capability of the proposed filters, the average quality metrics of the noisy images are also reported. From Table 2, it can be seen that, when σ_{noise}=5 (i.e. σr=15) and the kernel size is 5 × 5 (i.e. σs=1), the proposed approach improves the PSNR of noisy images by 2.88dB. With σ_{noise} equal to 20 and 30, when the 7 × 7 kernel size (i.e. σs=2) is adopted, such an improvement increases to 7.19dB and 8.29dB, respectively. Table 2 clearly shows that the denoising effects of the new filters are very close to, and sometime better than, those given by the MATLAB counterparts.

Figure 10 provides a compact overview in terms of relative errors introduced with respect to the accurate software MATLAB filters, as a function of σ_{noise} and k. The negative relative errors for the PSNR and the SSIM bounded to −4% and −14%, respectively, clearly demonstrate the graceful behavior of the proposed approach. It is also worth noting that, in several cases, the new filters introduce positive relative errors, which means that they perform even better than the accurate counterpart.

The above results also show that, as expected, the most convenient kernel size has to be chosen in relation to the specific application, that is to the expected σ_{noise} range. As an example, if the latter is between 5 and 30, the 11 × 11 kernel size seems to be the best choice, since it performs even better than its accurate counterpart. Conversely, the 3 × 3 kernel appears as the most favorable size when $\sigma_{noise} \geq 40$ is expected. However, in practical scenarios, the optimal kernel size choice must also consider available resources and power

TABLE 3. Implementation results and comparison with state-of-the-art counterparts.

Circuit	Device	$n \times m$	$k \times k$	σr	$PSNR_{dB}$/SSIM	#LUTs	#FFs	#BRAMs (18 Kb)	#DSPs	Freq. [MHz]	ppcc	fps	Mps	P [mW]
[15]	XC5VLX50	512×512	5×5	30	31.00 / 0.85[1]	n.a.	n.a.	11	29	220	1	209.8	52.45	n.a.
[17]	XC5VLX50	512×512	5×5	30	31.78 / 0.83[2]	5142	n.a.	0	36	236.7	1/4	225.7	56.43	n.a.
[19]	XC5VLX50	3268×2448	5×5	n.a.*	34.91 / 0.89[3]	2529	1917	2×4	27	242	1	30.3	230.8	n.a.
New^{Fix}	XC5VLX50	512×512	5×5	30	32.74 / 0.86[2]	2103	2217	0	24	239.8	1	914.8	228.7	n.a.
New^{Fix}	XC5VLX50	3268×2448	5×5	30	33.99 / 087[3]	1541	1691	2×4	24	243.9	1	30.5	232.6	n.a.
[16]	XC7Z020	256×256	4×4	90	26.97 / 0.48[4]	2585	686	157	10	62	1/9	103.3	6.45	n.a.
[18]	XC7V330T	1920×1080	5×5	n.a.	n.c.[5]	1425	552	8	8	330	1	159.2	314.7	n.a.
[19]	Zynq7000	3268×2448	5×5	n.a.	34.91 / 0.89[3]	2391	1812	2×4	27	278	1	34	259.4	168
New^{Fix}	XC7Z020	256×256	5×5	90	26.45 / 0.54[4]	1625	2379	0	32	244	1	3703.7	231.5	57
New^{Fix}	XC7Z020	512×512	5×5	30	32.74 / 0.86[2]	1689	1635	0	32	244	1	926.8	231.5	63
New^{Fix}	XC7Z020	512×512	5×5	90	26.76 / 0.65[2]	1881	2635	0	32	244	1	926.8	231.5	60
New^{Fix}	XC7Z020	1024×1024	5×5	90	26.60 / 0.55[6]	1357	2118	4	32	238	1	226.7	226.7	88
New^{Fix}	XC7Z020	3268×2448	5×5	90	33.99 / 0.87[3]	1357	2120	2×4	32	238	1	29.3	226.7	91
New^{Rec}	XC7Z020	512×512	5×5	15 ÷ 225	36.73/0.93 ÷ 19.85/0.28[7]	2487	2587	0	32	209	1	794.7	198.7	122
New^{Rec}	XC7Z020	1024×1024	5×5	15 ÷ 225	36.16/0.91 ÷ 19.82/0.26[6]	1886	2046	4	32	209	1	198.7	198.7	154
New^{Rec}	XC7Z020	3268×2448	5×5	15 ÷ 225	37.10/0.94 ÷ 19.73/0.29[8]	1921	2040	2×4	32	209	1	26.04	198.7	171

* Not available
[1] Results provided in [15] are related just to the test image 512×512 Lighthouse.
[2] Average PSNR/SSIM related to the 512×512 test images Barbara, Boat, Bridge, Couple, Goldhill, Lake, Lena, Lighthouse, Peppers and Plane.
[3] Results related just to the 512×512 test image Lena with σr=30.
[4] Results related just to the 256×256 test image Lena.
[5] Not comparable since related to unavailable test images on-purpose captured and processed in [18].
[6] Average PSNR/SSIM related to the 1024×1024 test images available in the Miscellaneous USC-SIPI dataset [22].
[7] Average PSNR/SSIM related to the 512×512 test images available in the Miscellaneous USC-SIPI dataset [22].
[8] Average PSNR/SSIM related to the entire Miscellaneous USC-SIPI dataset [22].

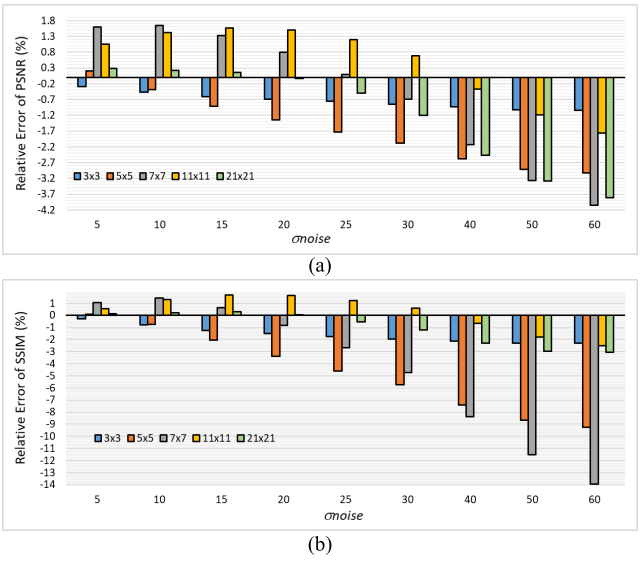

FIGURE 10. Relative error on the average: (a) PSNR; (b) SSIM.

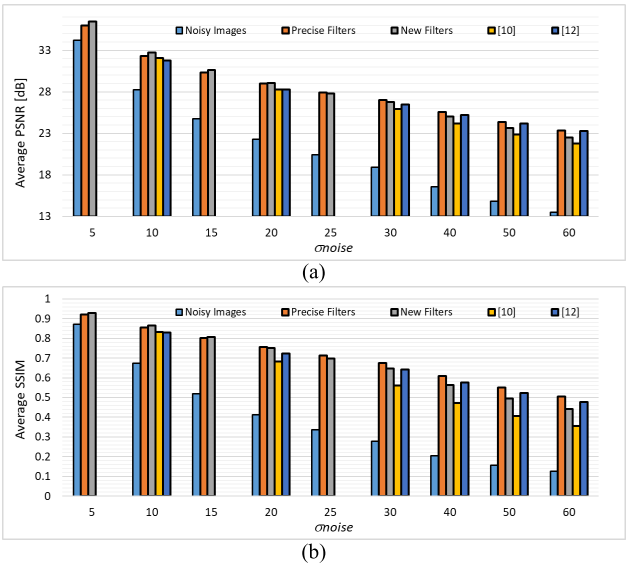

FIGURE 11. Average quality metrics on ten tests images: (a) PSNR; (b) SSIM.

budget. This aspect will be discussed more in detail in the next sub-Section.

For purposes of comparison with existing approaches designed to work at fixed σr, quality tests have been done also on the reduced data set used in [15] and [17], including the ten test images Barbara, Boat, Bridge, Couple, Goldhill, Lake, Lena, Lighthouse, Peppers and Plane. Results plotted in Figure 11 demonstrate that the new σr-adaptive approximation strategy does not introduce any quality penalty with respect to [17] and systematically overcomes [15].

Finally, Figure 12 reports sample images corrupted with several additive noise levels and allows comparing the images filtered using the MATLAB accurate routine with those obtained by the proposed approach. Denoising effects are more evident in Figures 13, 14 and 15 that illustrate some zoomed in details of the benchmark images.

B. IMPLEMENTATION RESULTS

For a fair comparison with hardware designs characterized at fixed values of σr, the architecture above shown in Figure 6 has been implemented to denoise digital images with 5×5 kernels at $\sigma s = 1$. Several versions of the proposed filters have been customized to work on predefined values of

FIGURE 12. Sample images: (a) corrupted with $\sigma noise = 15$; (b) filtered by MATLAB routine with $\sigma r = 45$; (c) filtered applying the new approach with $\sigma rfix = 60$ and $\sigma r = 45$; (d) corrupted with $\sigma noise = 30$; (e) filtered by MATLAB routine with $\sigma r = 90$; (f) filtered applying the new approach with $\sigma rfix = 60$ and $\sigma r = 90$; (g) corrupted with $\sigma noise = 75$; (h) filtered by MATLAB routine with $\sigma r = 225$; (i) filtered applying the new approach with $\sigma rfix = 60$ and $\sigma r = 225$.

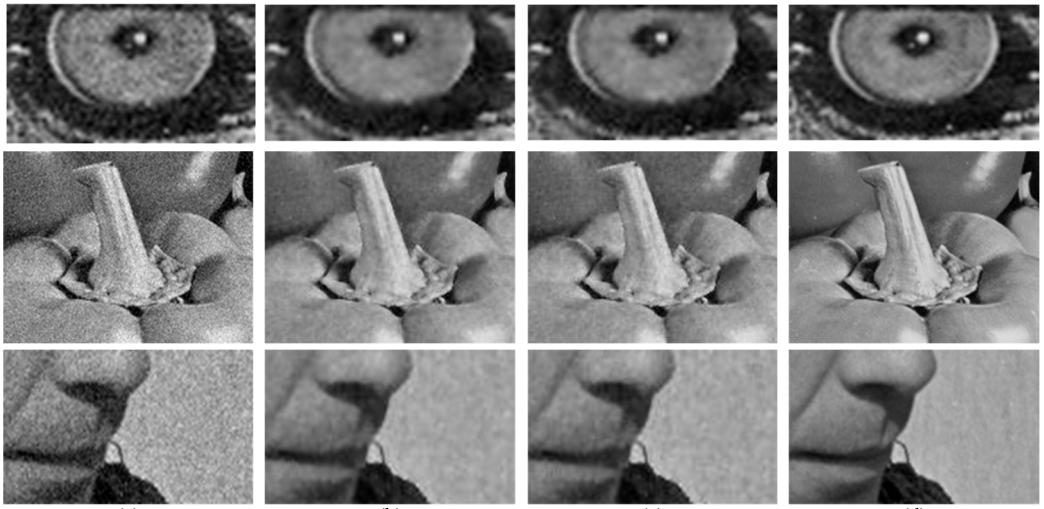

FIGURE 13. Details of sample images: (a) corrupted with $\sigma noise = 15$; (b) filtered by MATLAB routine with $\sigma r = 45$; (c) filtered applying the new approach with $\sigma rfix = 60$ and $\sigma r = 45$; (d) original noise-free.

σr. Then, the run-time reconfigurable design has been carried out using $\sigma rfix = 60$. In the following, such designs are named New$^{\text{Fix}}$ and New$^{\text{Rec}}$, respectively.

All the circuits have been implemented using the Xilinx Zynq-7000 device family and the Vivado 2019.2 Software Development Tool. Just for purpose of comparison with previous designs [15], [16], [17], [19], the proposed architecture has also been synthesized on the obsolete Virtex-5 hardware platform by using the ISE 14.7 Design Suite on a Virtual Machine Oracle VM.

Table 3 summarizes implementation results achieved in terms of: quality performances, hardware resources utilization, as the amount of occupied LUTs, FFs, BRAMs, and DSPs; speed performances, evaluated in terms of maximum running frequency, number of filtered pixels outputted per clock cycle (ppcc), number of frame and Mega pixels processed per second (fps and Mps); power consumption, analyzed using the interchange format files (SAIF) extracted from post-implementation simulations performed on test images. It can be seen that, at a parity of σr, images and kernels sizes, the New$^{\text{Fix}}$ designs occupy up to 70% and 12% less LUTs and FFs than their direct competitors. It is also worth observing that those new implementations with an increased amount of occupied LUTs and/or FFs significantly reduces the utilized BRAMs, and vice versa.

In terms of speed performances, the New$^{\text{Fix}}$ implementations reach frame rates significantly better than their counterparts and exhibit pixels rates $\times 4.4$, $\times 35$ and $\times 4.1$ times higher than [15], [16], and [17], respectively. In comparison with [19], the new design is $\sim$12.6% slower, but, on the other hand, it saves more than 45% of the consumed power. Such an advantage is obtained without significantly affecting the PSNR/SSIM metrics that are less than 0.3% lower. Table 3 confirms the nice behavior of the New$^{\text{Fix}}$ designs in terms of quality performances also in comparison with [15], [16], and [17].

As expected, due to the more complex architecture of the employed PEs, the New$^{\text{Rec}}$ designs utilize more hardware

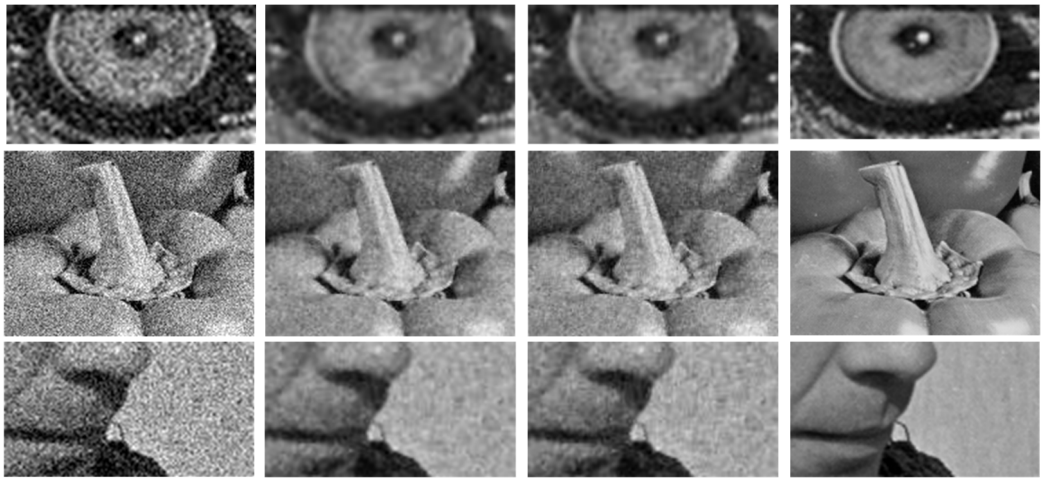

FIGURE 14. Details of sample images: (a) corrupted with $\sigma noise = 30$; (b) filtered by MATLAB routine with $\sigma r = 90$; (c) filtered applying the new approach with $\sigma rfix = 60$ and $\sigma r = 90$; (d) original noise-free.

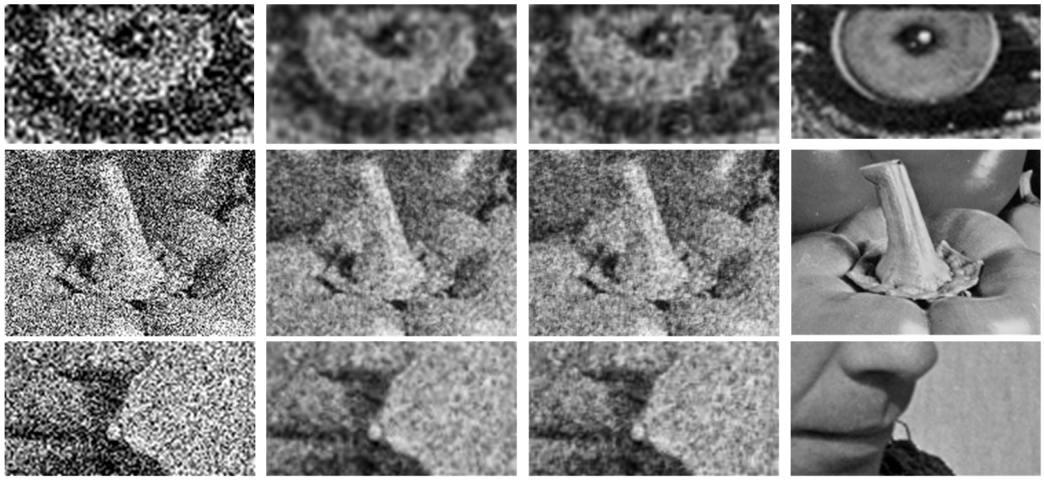

FIGURE 15. Details of sample images: (a) corrupted with $\sigma noise = 75$; (b) filtered by MATLAB routine with $\sigma r = 225$; (c) filtered applying the new approach with $\sigma rfix = 60$ and $\sigma r = 225$; (d) original noise-free.

resources than the New^{Fix} implementations and achieve somewhat slower computational speeds. This is the more than reasonable price to pay for obtaining the capability of run-time adapting the denoising filtering to variable noise levels. It is worth underlining that most of the referred competitors does not have this property. This means that, in the presence of variable noise levels such designs cannot guarantee the same quality performances reported in Table 3. As an alternative, the used device could be reconfigured to upload the properly customized design complying with a different value of σr. However, the not negligible reconfiguration time [24] makes this solution unsuitable to fast adapt the filtering capability to run-time varying noise levels. Conversely, the new approach makes the proposed designs able to run-time adapt them-selves to various noise levels without compromising either the quality performances or the overall computational speeds.

As concluding remarks, it is worth noting that the amount of hardware resources occupied by the above analyzed design, based on the 5×5 kernel, is $1.76\times$ and $4.16\times$ lower than those required by the implementations adopting 7×7 and 11×11 kernels, respectively. Moreover, it guarantees an average PSNR higher than 30dB on the entire reference dataset [22], with σ_{noise} ranging between 5 and 30. For these reasons, such configuration could represent the best tradeoff for a wide range of practical applications.

VI. CONCLUSION

This paper presented an innovative approximation strategy for the hardware implementation of fast low-cost denoising filters based on a new simple piecewise function. In comparison with the accurate software filters, the accuracy loss introduced by the novel approach in terms of PSNR and SSIM is bounded to just 4% and 14%, respectively. In addition to many positive aspects, the proposed system also inherits the limitations of the bilateral filter approach, such as the tendency to remove most of the texture and fine details, or the creation of flat intensity regions and new contours.

Hardware architectures designed as proposed here are able to support different noise levels varying at run-time and overcome state-of-the-art counterparts in terms of both accuracy metrics and hardware characteristics. At a parity of the target FPGA devices, images and kernel sizes, as well as noise levels, the new filters exhibited pixels rates up to $\times 35$ times higher than competitors with power saving up to more than 45%.

REFERENCES

[1] (Jan. 2021). *European Commission, Industry 5.0: Towards a Sustainable, Human-Centric and Resilient European Industry*. [Online]. Available: https://op.europa.eu/en/publication-detail/-/publication/468a892a-5097-11eb-b59f-01aa75ed71a1/

[2] P. K. R. Maddikunta, Q.-V. Pham, B. Prabadevi, N. Deepa, and K. Dev, "Industry 5.0: A survey on enabling technologies and potential applications," *J. Ind. Inf. Integr.*, vol. 26, pp. 1–31, Mar. 2022, doi: 10.1016/j.jii.2021.100257.

[3] J. Nakamura, *Image Sensors and Signal Processing for Digital Still Cameras*. New York, NY, USA: Taylor & Francis, 2006, ch. 3, pp. 55–90.

[4] D.-H. Shin, R.-H. Park, S. Yang, and J.-H. Jung, "Block-based noise estimation using adaptive Gaussian filtering," *IEEE Trans. Consum. Electron.*, vol. 51, no. 1, pp. 218–226, Feb. 2005, doi: 10.1109/TCE.2005.1405723.

[5] C. Liu, W. T. Freeman, R. Szeliski, and S. B. Kang, "Noise estimation from a single image," in *Proc. IEEE CVPR*, Jun. 2006, pp. 901–908, doi: 10.1109/CVPR.2006.207.

[6] G. Chen, F. Zhu, and P. A. Heng, "An efficient statistical method for image noise level estimation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 477–485, doi: 10.1109/ICCV.2015.62.

[7] S. Prasath and D. N. Thanh, "Structure tensor adaptive total variation for image restoration," *Turkish J. Electr. Eng. Comput. Sci.*, vol. 27, pp. 1147–1156, Mar. 2019, doi: 10.3906/elk-1802-76.

[8] D. N. H. Thanh, V. B. S. Prasath, L. M. Hieu, and S. Dvoenko, "An adaptive method for image restoration based on high-order total variation and inverse gradient," *Signal, Image Video Process.*, vol. 14, no. 6, pp. 1189–1197, Feb. 2020, doi: 10.1007/s11760-020-01657-9.

[9] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proc. 6th Int. Conf. Comput. Vis.*, Jan. 1998, pp. 839–845, doi: 10.1109/ICCV.1998.710815.

[10] B. K. S. Kumar, "Image fusion based on pixel significance using cross bilateral filter," *Signal, Image Video Process.*, vol. 9, no. 5, pp. 1193–1204, 2015, doi: 10.1007/s11760-013-0556-9.

[11] B. Kaur, A. Dogra, and B. Goyal, "Comparative analysis of bilateral filter and its variants for magnetic resonance imaging," *Open Neuroimaging J.*, vol. 13, no. 1, pp. 21–29, Nov. 2020, doi: 10.2174/1874440002013010021.

[12] B. K. Gunturk, "Fast bilateral filter with arbitrary range and domain kernels," *IEEE Trans. Image Process.*, vol. 20, no. 9, pp. 2690–2696, Sep. 2011, doi: 10.1109/TIP.2011.2126585.

[13] S. Ghosh, P. Nair, and K. N. Chaudhury, "Optimized Fourier bilateral filtering," *IEEE Signal Process. Lett.*, vol. 25, no. 10, pp. 1555–1559, Oct. 2018, doi: 10.1109/LSP.2018.2866949.

[14] B.-H. Chen, Y.-S. Tseng, and J.-L. Yin, "Gaussian-adaptive bilateral filter," *IEEE Signal Process. Lett.*, vol. 27, pp. 1670–1674, Sep. 2020, doi: 10.1109/LSP.2020.3024990.

[15] A. Gabiger-Rose, M. Kube, R. Weigel, and R. Rose, "An FPGA-based fully synchronized design of a bilateral filter for real-time image denoising," *IEEE Trans. Ind. Electron.*, vol. 61, no. 8, pp. 4093–4104, Aug. 2014, doi: 10.1109/TIE.2013.2284133.

[16] S. D. Dabhade, G. N. Rathna, and K. N. Chaudhury, "A reconfigurable and scalable FPGA architecture for bilateral filtering," *IEEE Trans. Ind. Electron.*, vol. 65, no. 2, pp. 1459–1469, Feb. 2018, doi: 10.1109/TIE.2017.2726960.

[17] C.-Y. Lien, C.-H. Tang, P.-Y. Chen, Y.-T. Kuo, and Y.-L. Deng, "A low-cost VLSI architecture of the bilateral filter for real-time image denoising," *IEEE Access*, vol. 8, pp. 64278–64283, 2020, doi: 10.1109/ACCESS.2020.2984688.

[18] S.-J. Jang and Y. Hwang, "Noise-aware and light-weight VLSI design of bilateral filter for robust and fast image denoising in mobile systems," *Sensors*, vol. 20, pp. 1–15, Aug. 2020, doi: 10.3390/s20174722.

[19] R. Yao, L. Chen, P. Dong, Z. Chen, and F. An, "A compact hardware architecture for bilateral filter with the combination of approximate computing and look-up table," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 69, no. 7, pp. 3324–3328, Jul. 2022, doi: 10.1109/TCSII.2022.3159261.

[20] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004, doi: 10.1109/TIP.2003.819861.

[21] A. Hore and D. Ziou, "Image quality metrics: PSNR vs. SSIM," in *Proc. Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2010, pp. 2366–2369, doi: 10.1109/ICPR.2010.579.

[22] *USC-SIPI Image Daset*. Accessed: Jan. 3, 2023. [Online]. Available: https://sipi.usc.edu/database/database.php?volume=misc

[23] J. E. Robertson, "A new class of digital division methods," *IRE Trans. Electron. Comput.*, vol. 7, no. 3, pp. 218–222, Sep. 1958, doi: 10.1109/TEC.1958.5222579.

[24] *7 Series FPGAs Configuration User Guide*. Accessed: Jan. 3, 2023. [Online]. Available: https://docs.xilinx.com/v/u/en-US/ug470_7Series_Config

FANNY SPAGNOLO (Member, IEEE) was born in Belvedere Marittimo, Cosenza, Italy, in April 1991. She received the master's degree in electronics engineering and the Ph.D. degree in information and communication technologies from the University of Calabria, Italy, in 2016 and 2019, respectively. She is currently working as an Assistant Professor with the University of Calabria. She has coauthored of more than 30 papers in the field of VLSI design. Her research interests include VLSI architectures for image processing, high-performance reconfigurable circuits, embedded systems design, emerging technologies, and approximate computing techniques for low-power deep neural networks. In June 2016, she won a Research Grant funded by the Department of Informatics, Modeling, Electronics and System Engineering, University of Calabria.

PASQUALE CORSONELLO (Member, IEEE) was born in Cosenza, Italy, in May 1964. He received the master's degree in electronics engineering from the University of Naples Federico II, Naples, Italy, in 1988. He joined the Institute of Research on Parallel Computers, National Council of Research of Italy, Naples, where he was working on the design and modeling of electronic transducers for high precision measurement, receiving a postgraduate two-year grant. In 1992, he joined the Department of Electronics, Computer Science and Systems, University of Calabria, Rende, Italy, as a Research Associate. In 1997, he was appointed as an Assistant Professor of electronics with the Department of Electronics Engineering and Applied Mathematics, University of Reggio Calabria, Reggio Calabria, Italy, where he also worked as the Director of the Microelectronics Laboratory. In 2001, he was appointed as an Associate Professor of electronics and as the Chair of the Ph.D. Program in Electronics Engineering at the University of Reggio Calabria. In 2004, he joined the Department of Electronics Computer Science and Systems, University of Calabria. In Summer 2004, he was a Visiting Researcher at the Department of Electrical and Computer Engineering, University of Rochester, Rochester, NY, USA. In 2005, he was appointed as an Adjunct Associate Professor at the Department of Electrical and Computer Engineering. In 2021, he was appointed as a Full Professor of electronics at the Department of Informatics, Modeling, Electronics and System Engineering, University of Calabria. He has coauthored over 180 technical papers and holds two patents in these fields. His main research interests include embedded systems design, low-power design, VLSI architecture for image processing, and QCA-based circuits. He serves on technical committees for several VLSI conferences and as a peer reviewer for several VLSI journals. He served as the Editor-in-Chief for the *Journal of Low Power Electronics and Applications*. He is also an Associate Editor-in-Chief of the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION VLSI SYSTEMS.

FABIO FRUSTACI (Senior Member, IEEE) was born in Italy, in 1979. He received the M.S. and Ph.D. degrees in electronic engineering from the Mediterranea University of Reggio Calabria, Italy, in 2003 and 2007, respectively. In 2006, he was a Visiting Research Associate at the Department of ECE, University of Rochester, Rochester, NY, USA. From 2011 to 2013, he was a Visiting Researcher at the Department of EECS, University of Michigan, Ann Arbor, MI, USA. In 2014, he joined the DIMES, University of Calabria, Italy, as an Assistant Professor, where he is currently an Associate Professor. He has authored more than 60 papers in the field of VLSI design. His research interests include ultra low power and high-performance design, VLSI circuits, design techniques for emerging technologies, reconfigurable architectures, and embedded systems. He was a member of the Technical Program Committee of several conferences. He is also a member of the Editorial Board of *Microelectronics Journal*.

STEFANIA PERRI (Senior Member, IEEE) was born in Cosenza, Italy, in April 1971. She received the master's degree in computer science engineering from the University of Calabria, Italy, in 1996, and the Ph.D. degree in electronics engineering from the Mediterranea University of Reggio Calabria, Italy, in 2000. In 1996, she joined the Department of Electronics, Computer Sciences and Systems, University of Calabria, as a Researcher Associate. In 2002, she was appointed as an Assistant Professor of electronics at the Department of Electronics, Computer Science and Systems, University of Calabria. In Summer 2004, she was a Visiting Researcher at the Department of Electrical and Computer Engineering, University of Rochester, Rochester, NY, USA, where she was appointed as an Adjunct Assistant Professor for four years, in 2005. In 2010, she was appointed as an Associate Professor of electronics at the Department of Electronics, Computer Sciences and Systems, University of Calabria. In 2017, she joined the Department of Mechanical, Energy and Management Engineering, University of Calabria. Her current research interests include QCA-based circuits, high-performance embedded systems, low-power design, VLSI circuits for image processing and multimedia, reconfigurable computing, and VLSI design. She is the coauthor of more than 140 technical papers and holds two patents in these fields. She serves on technical committees for several VLSI conferences and as a peer reviewer for several VLSI journals. She is currently an Associate Editor of the *Journal of Low Power Electronics and Applications* and *Sensors*.

Open Access funding provided by 'Università della Calabria' within the CRUI CARE Agreement